Java RMI (Remote Method Invocation)

Tanja Mustajoki

Avoimien hajautettujen järjestelmien tutkimusseminaari
Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Helsinssä 7.2.2000

Java RMI (*Remote Method Invocation*) on perinteisen Sun:n kehittämän RPC:n *(Remote Procedure Call*) kaltainen protokolla, jonka avulla eri koneilla olevat ohjelmat kommunikoivat keskenään. Suurimpana erona RPC:hen on se, että RMI:tä voidaan käyttää vain Java-ohjelmien väliseen kommunikointiin perinteisen RPC:n soveltuessa kaikkiin ohjelmointikieliin. RMI:n Java-pohjaisuus mahdollistaa ohjelman helpon ohjelmoinnin ja yksinkertaiset rajapinnat. RMI:n avulla kutsussa voidaan välittää parametreina ja paluuarvoina kokonaisia olioita perinteisten hyvin yksinkertaisten tietorakenteiden sijaan.

RMI on siis palvelu, jonka avulla yhdellä koneella oleva Java-ohjelma voi herättää toisella koneella olevan objektin metodin. Kommunikointi perustuu yleensä asiakas-palvelin -malliin. Asiakas kutsuu palvelijan tarjoamia palveluja annettujen metodikutsujen välityksellä.

Java on erityisen turvallinen ohjelmointiympäristö ja sen takia myös RMI:tä pidetään turvallisena kommunikointitapana. Palvelijalla oleva `SecurityManager` objekti tarkistaa, että kaikki käytettävät operaatiot ovat sallittuja. Eri käyttäjäryhmille on mahdollista antaa erilaisia oikeuksia suorittaa metodeita. Esimerkiksi ulkopuolisille palvelun käyttäjille voidaan antaa mahdollisuus lukea yrityksen tiettyjä tietokantoja, mutta tietojen muuttamismahdollisuus annetaan vain yrityksen sisäisille henkilöille.

Normaalissa Java-metodikutsussa välitettävää objektia ei siirretä kutsutun metodin muistialueelle, vaan kutsussa välitetään vain viite kyseiseen objektiin. Tälläisenaan objektin välitys ei toimi hajautetussa kommunikoinnissa. RMI:n metodikutsussa on joko välitettävä erityinen kaukoviite (*remote reference*, viite, joka osoittaa paikallisen koneen muistissa olevaan objektiin) tai kopioitava koko objekti palvelimelle käytettäväksi. Objektia kopioitaessa paikalliselta koneelta etäkoneelle on objekti muutettava tavuvirraksi (*a stream of bytes*). Koska objekti voi sisältää sisäisiä objekteja, tämä operaatio muodostuu vähintäänkin monimutkaiseksi.

RMI:n suuria heikkouksia on sen rajoittuminen Javaan. RMI:n avulla ei voi välittää kuin Javalla kirjoitettuja metodeja. RPC on yksi keino tämän ongelman ratkaisuuun, mutta se soveltuu huonosti olioperusteiseen ohjelmaan. Hajautettuja olioita varten on kehitetty CORBA (*Common Request Broker Architecture*), joka mahdollistaa eri ohjelmointikielillä kirjoitettujen objektien kommunikoinnin. Java linkittyy CORBA:aan IDL:n (*Java Interface Definition Language*) ja JOE:n (*Java Object Environment*) avulla.

Neljä tärkeintä pakettia, joiden metodeita tarvitaan RMI:n toteutuksessa, ovat: `java.rmi`, `java.rmi.server`, `java.rmi.registry` ja `java.rmi.dgc`. `Java.rmi`-paketti

määrittelee luokat, rajapinnat ja poikkeukset, jotka näkyvät asiakkaan puolella ja `Java.rmi.server`-paketti samat luokat palvelimen puolella. `Java.rmi.registry`-paketti taas määrittelee luokat, rajapinnat ja poikkeukset, joiden avulla nimetään ja paikannetaan palvelijan objektit. `Java.rmi.dgc`-paketti huolehtii hajautetusta roskien keruusta.

Jotta saavutetaan yhteensopivuus eri Java-ohjelmien välillä, ja jotta tiedonsiirto on mahdollisimman tuntumatonta ohjelmoijalle, kommunikointi asiakkaan ja palvelimen välillä toteutetaan useamman kerroksen avulla. Nämä kerrokset ovat loogisen yhteyden muodostava ohjelmakerros (*Client/Server Program*), luuranko-ohjelma (*skeleton*), viitekerros (*remote reference layer*) ja itse kuljetuksesta huolehtiva kuljetukerros (*Transport Layer*).

Lähteet:

[Dow98] Downing, T.B., Java RMI: remote method invocation, 1998, IDG Books Worldwide, Inc., USA 1998.

[Har97] Harold, E.R., Java network programming. O'Reilly & Associates, Inc., USA 1997, 347-374.